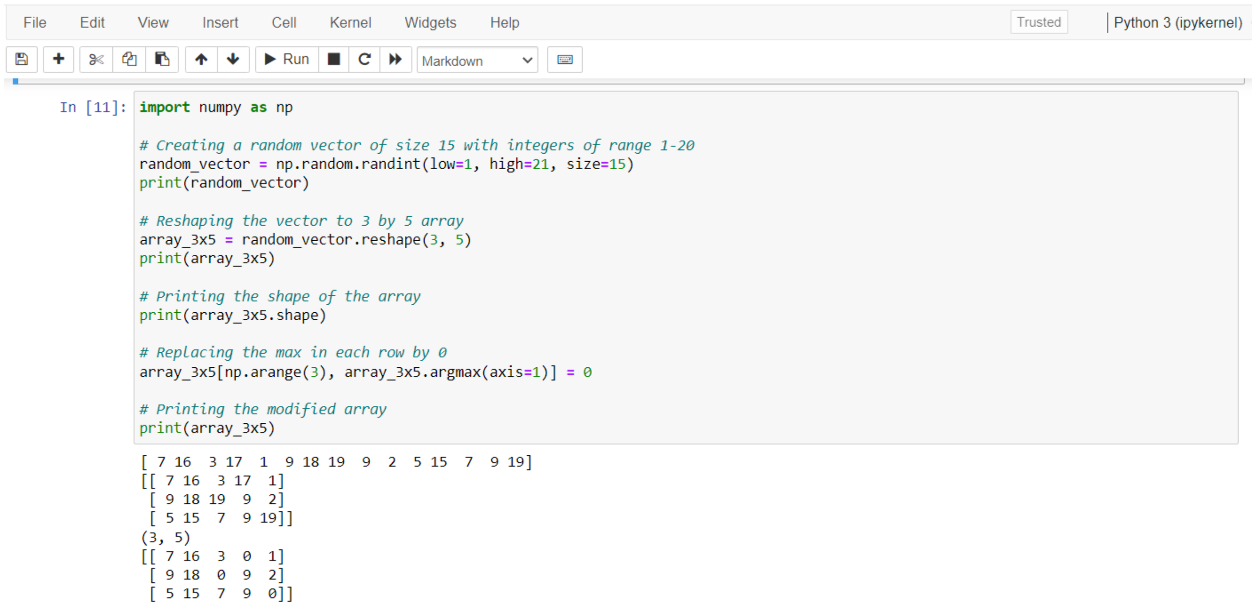
Machine Learning ICP3

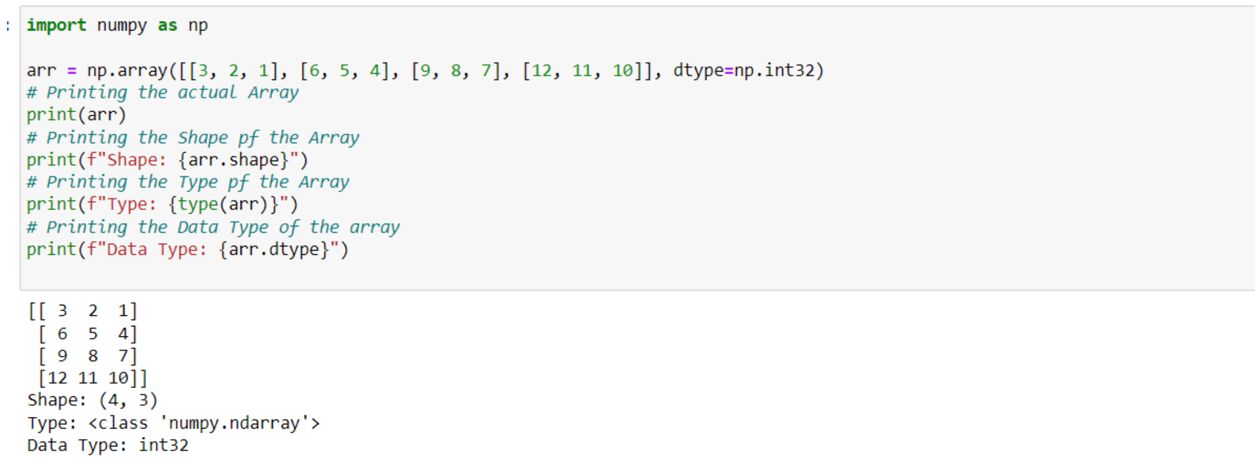
Q1:



```
In [11]: import numpy as np

         # Creating a random vector of size 15 with integers of range 1-20
         random_vector = np.random.randint(low=1, high=21, size=15)
         print(random_vector)

         # Reshaping the vector to 3 by 5 array
         array_3x5 = random_vector.reshape(3, 5)
         print(array_3x5)

         # Printing the shape of the array
         print(array_3x5.shape)

         # Replacing the max in each row by 0
         array_3x5[np.arange(3), array_3x5.argmax(axis=1)] = 0

         # Printing the modified array
         print(array_3x5)
```

```
[ 7 16  3 17  1  9 18 19  9  2  5 15  7  9 19]
[[ 7 16  3 17  1]
 [ 9 18 19  9  2]
 [ 5 15  7  9 19]]
(3, 5)
[[ 7 16  3  0  1]
 [ 9 18  0  9  2]
 [ 5 15  7  9  0]]
```

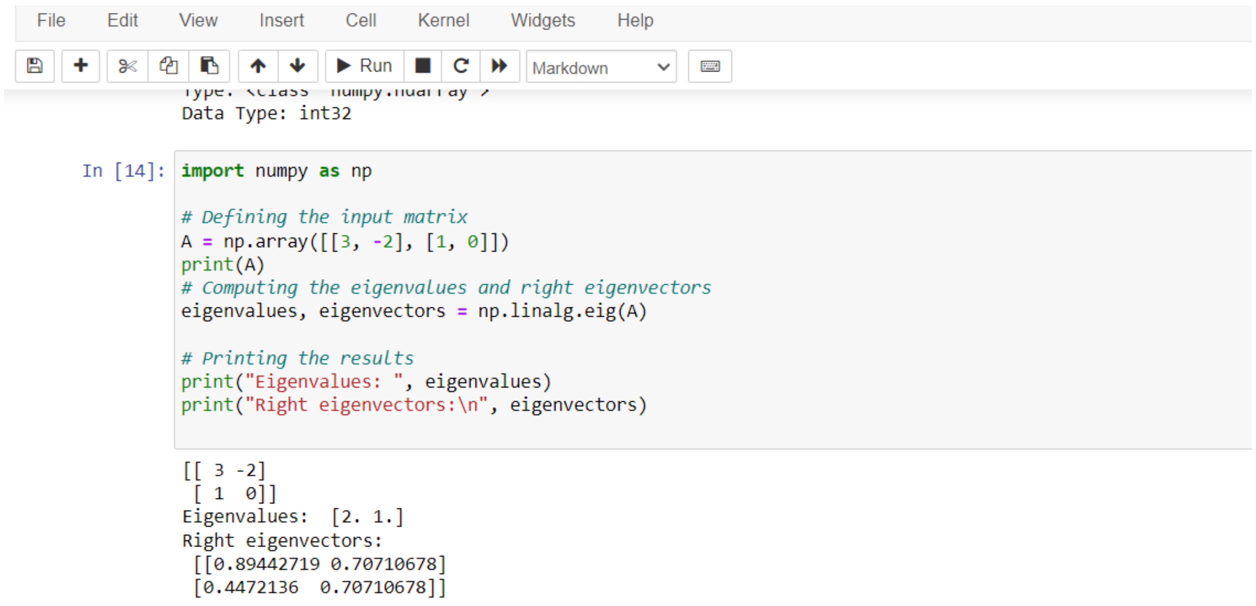
Q2:



```
import numpy as np

arr = np.array([[3, 2, 1], [6, 5, 4], [9, 8, 7], [12, 11, 10]], dtype=np.int32)
# Printing the actual Array
print(arr)
# Printing the Shape pf the Array
print(f"Shape: {arr.shape}")
# Printing the Type pf the Array
print(f"Type: {type(arr)}")
# Printing the Data Type of the array
print(f"Data Type: {arr.dtype}")
```

```
[[ 3  2  1]
 [ 6  5  4]
 [ 9  8  7]
 [12 11 10]]
Shape: (4, 3)
Type: <class 'numpy.ndarray'>
Data Type: int32
```

Q3:

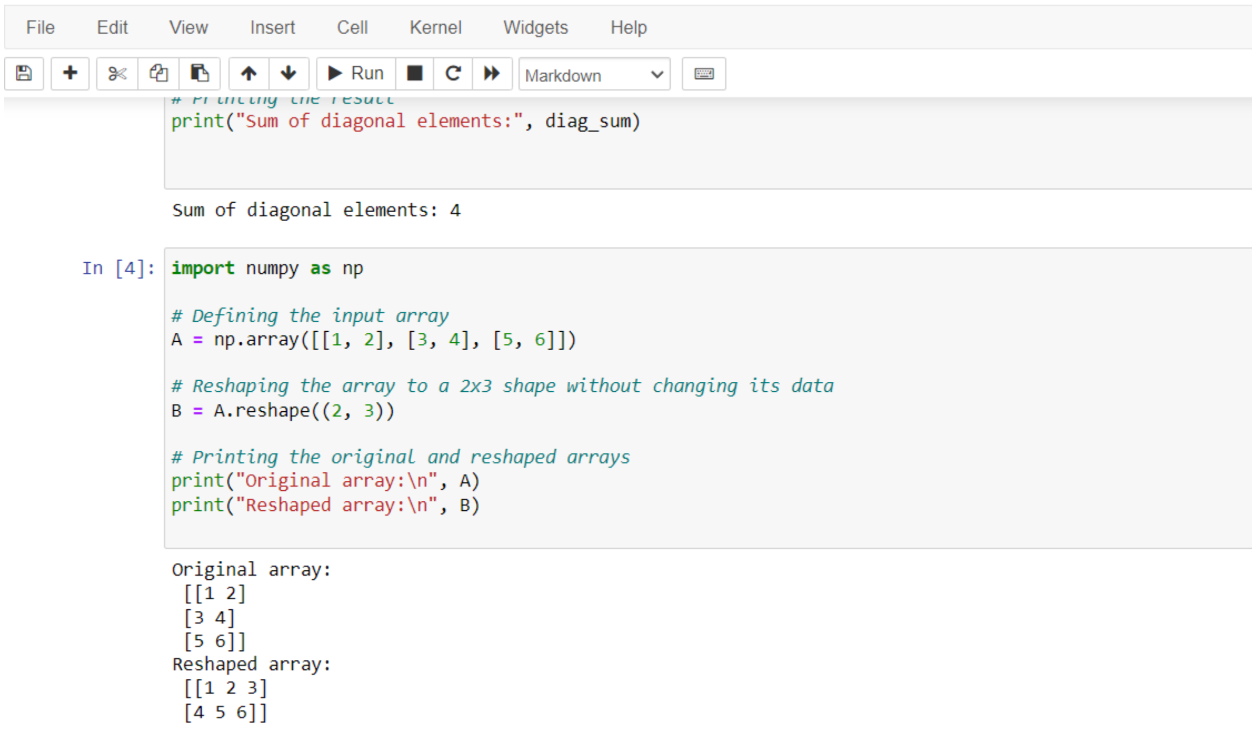


```
Type: <class 'numpy.ndarray'>
Data Type: int32
```

In [14]:
```
import numpy as np

# Defining the input matrix
A = np.array([[3, -2], [1, 0]])
print(A)
# Computing the eigenvalues and right eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

# Printing the results
print("Eigenvalues: ", eigenvalues)
print("Right eigenvectors:\n", eigenvectors)
```

```
[[ 3 -2]
 [ 1  0]]
Eigenvalues:  [2. 1.]
Right eigenvectors:
 [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

Q4:



```
# Printing the result
print("Sum of diagonal elements:", diag_sum)
```

```
Sum of diagonal elements: 4
```

In [4]:
```
import numpy as np

# Defining the input array
A = np.array([[1, 2], [3, 4], [5, 6]])

# Reshaping the array to a 2x3 shape without changing its data
B = A.reshape((2, 3))

# Printing the original and reshaped arrays
print("Original array:\n", A)
print("Reshaped array:\n", B)
```

```
Original array:
 [[1 2]
 [3 4]
 [5 6]]
Reshaped array:
 [[1 2 3]
 [4 5 6]]
```